N88-29380

ARGES: An Expert System for Fault Diagnosis within Space-Based ECLS Systems

David W. Pachura
Salem A. Suleiman
Andrew P. Mendler

Martin Marietta Denver Aerospace
Space Station Program
P. O. Box 179 (MS D1744)
Denver, CO 80201

Abstract

ARGES (the Atmosphere Revitalization Group Expert System) is a demonstration prototype expert system for fault management for the Solid Amine, Water Desorbed (SAWD) CO_2 removal assembly, aassociated with the Environmental Control and Life Support (ECLS) System. ARGES monitors and reduces data in real time from either the SAWD controller or a simulation of the SAWD assembly. It can detect gradual degradations or predict failures. This allows graceful shutdown and scheduled maintenance, which reduces crew maintenance overhead. Status and fault information is presented in a user interface that simulates what would be seen by a crewperson. The user interface employs animated color graphics and an object-oriented approach to provide detailed status information, fault identification, and explanation of reasoning in a rapidly assimilated manner. In addition, ARGES recommends possible courses of action for predicted and actual faults. ARGES is seen as a forerunner of AI-based fault managment systems for manned space systems.

Introduction

ARGES (the Atmosphere Revitalization Group Expert System) is the result of an independent research and development project (D-47s) at Martin Marietta to demonstrate the application of artificial intelligence to fault diagnosis and management in space-based Environmental Control and Life Support ECLS systems. The work was performed in conjuction with Hamilton Standard, Inc ,Windsor Locks ,Conn., who provided expert engineering and design knowledge regarding operations of the ECLS system assembly hardware/software. The goal was to show an increased flexibility and function within this task, providing greater assistance to the crew and reducing the need for ground-based support. The first phase of the development was the design and implementation of a prototype that performs fault detection and isolation and demonstrates the user interaction and interface with the overall management software. In this paper, we discuss some of the significant features of ARGES, the architecture and current state of the system, and conclude with some areas of future work.

Approach

ARGES is an expert system for fault diagnosis of the Solid Amine Water Desorbed (SAWD) CO2 Removal Assembly. It is a prototype for demonstrating the applicability of AI/Expert Systems technology to space-based ECLS systems and its function as part of the overall management software system. With that goal in mind, the resulting system departs from other work previously done in the ECLS system area [Dickey84, Lance85]. Its most important features are:

1. It is a prototype of an expert system which functions as part of the control and management software for the ECLSS; not as an isolated system which communicates with a user, but as an embedded program, which uses data received directly from the hardware. It interprets the data, detects a fault, and reaches a conclusion without human intervention. Any interaction between the user and the program occurs at the user's option and convenience, as a means of verifying the conclusion, not as an aid to the program's diagnosis.

2. The user interface is designed to simulate what would be seen by an on-board crewmember — we have simulated enough of interface to see how an expert system could interact with other components of the manager, and to see how a sophisticated user interface can support a crewmember (see [Greitzer86]).

3. A major goal of ARGES is to recognize potential faults before they cause shutdown of the hardware. Currently, the alternatives to using an expert system are either to run the system until it fails and then diagnose the problem, or employ a ground-based human operator to monitor the hardware telemetry downlink for degradations. By adding a forecasting function which is not currently performed by the controller, we can improve crew utilization and reduce the "fire-fighting" mode of operation.

4. ARGES generates recommendations for action based on a simulation of the space environment. Thus , we are beginning to automate the massive operating procedures manuals currently in use, which detail all known contingencies and procedures for dealing with them. By providing this as part of the overall fault handling, crew training for contingencies can be reduced.

Description of the ARGES Problem Domain

The domain chosen for the prototype implementation was the Atmosphere Revitalization Subsystem within ECLSS. In particular, we focused on the atmosphere revitalization "group" of assemblies which remove CO2 from cabin air and replenish it with O2. In this group, there is a CO2 Removal Assembly, which removes CO2 from cabin air, a CO2 Reduction Assembly, which combines the CO2 with H2 to yield water plus either carbon or methane, and an O2 Generation Assembly which takes the water from the CO2 Reduction Assembly and generates pure O2 to be added to cabin air. For the prototype expert system, we focused on the CO2 Removal Assembly because it is the most complex of the three. We chose the SAWD system because we had access to the experts (see [Bailey86]). Although ARGES is designed to perform fault diagnosis for the entire atmosphere revitalization group, only the SAWD data are considered because access to data from the other components was not initially provided.

The use of a SAWD CO2 removal system for manned space platforms has been discussed in detail ([Boehm82]). The following is a brief description of the operation of the SAWD system to familiarize the reader with the technology.

The system consists of two canisters, or beds, of solid diethylenetriamine — "amine" — in a polystyrene substrate. During normal operation, one bed adsorbs CO2 from cabin air, while the other desorbs CO2. A fan draws air through the moist adsorbing amine bed. This cools and dries the bed, collecting CO2 on the amine particles in the bed. Steam is driven through the other bed to desorb it. This initially pushes the remainder of the purified cabin air, or ullage air, out of the bed and concentrates CO2 at one end of the bed. A sharp increase of flow out of the amine bed signals that CO2 is now being driven out of the bed. This causes a valve to switch and the CO2 to be directed to an accumulator and the CO2 reduction unit. An increase in the bed outlet temperature signals that all the CO2 has been driven off and the flow is now almost all steam. At this point, valves are switched to connect the output of the desorbing bed to the input of the adsorbing bed, so that the steam in the bed just desorbed can be used to heat the bed that was adsorbing. After this, the roles of the beds are reversed; ie., the desorbed bed now adsorbs CO2 and vice versa. Operation of the system is performed by a microprocessor that communicates via an RS-232 port or a MIL-STD 1553 network to an external controller/display unit. Fault handling in the controller is confined to limit checking on critical temperatures and times in the system, with out-of-bounds conditions resulting in automatic shutdown and an error indication being sent to the controller/display unit.

The ARGES Architecture

ARGES is organized into four major components (see Figure 1): the input processor, the expert system, the display manager, and the user interface. ARGES can read data from several sources — a software simulation of the SAWD hardware, an RS-232 link to the SAWD controller, or a file (consisting of stored RS-232 data). These data are passed to the input processor (which performs data transformation for the expert system) and to the display system

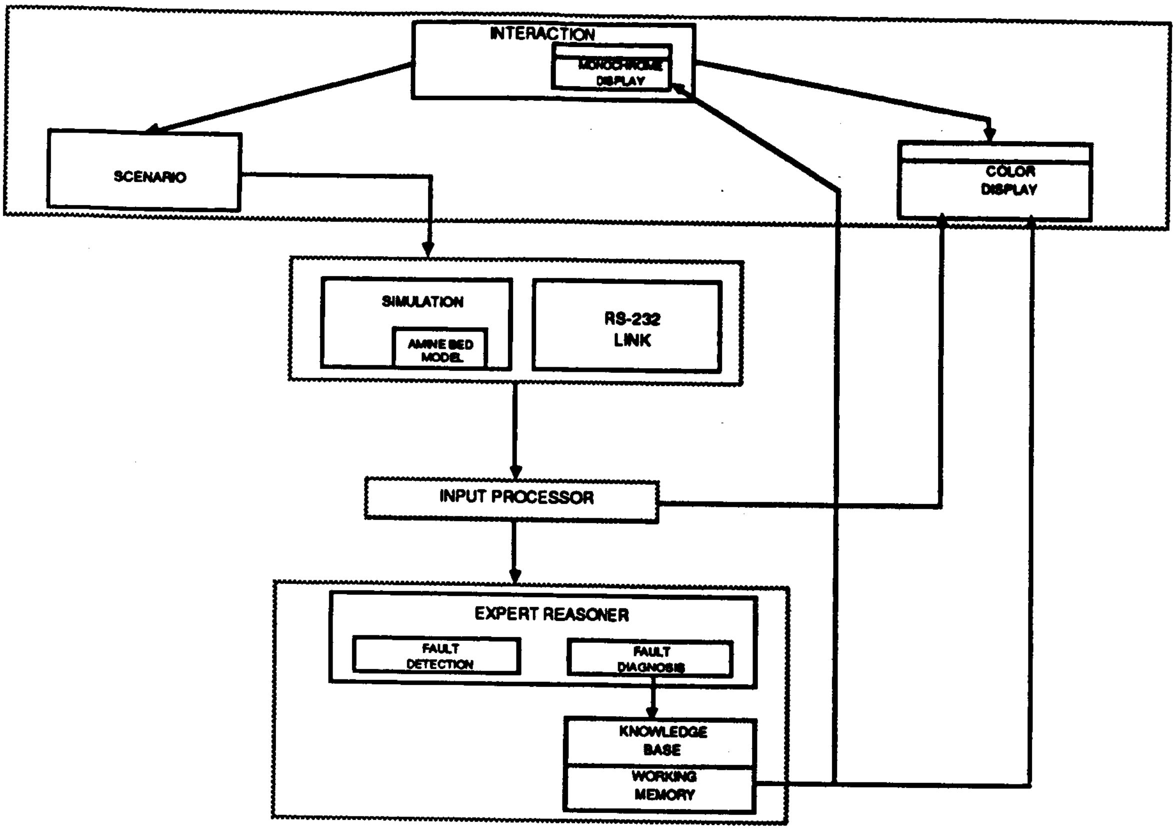

Figure 1. ARGES Structure

for updating the schematic display. After accumulating sufficient data, the expert system makes a diagnosis and sends the conclusion to the display system and user interface. Below, we discuss each of these components in more detail.

The Data Input Sources

The Simulation. The central part of the simulation is a simple model of the SAWD amine bed provided by Hamilton-Standard, Inc. This is a one node version of the computer model they use for hardware design, development and testing of the SAWD [Yanosy85]. The ARGES simulation of the SAWD system allows the graphics display and expert reasoner to be driven with reasonable accuracy in the absence of the actual hardware. Since the trending that the expert system performs may take several days, the use of the simulation at multiples of real-time allows testing more quickly than the actual hardware allows, even if we could induce faults in the hardware to test the system.

SAWD hardware via RS-232 link to controller. Because the SAWD operation cycle is on the order of two hours, and drifts and changes usually occur on the order of days,the only truly time-critical portion of ARGES is the link to the SAWD controller. Once a link is established, the input processor begins to receive information every several seconds. In order to handle these incoming data without lagging behind, the input reader and processor were implemented as processes to separate the real-time data processing from the more time-consuming, but less time-critical operations. Each process can be assigned a separate priority that determines the amount of processor time it receives. Currently in ARGES, the input processing does not operate at high priority, but if the amount of computation changes (through representation changes, or extensions to the expert system or interface), we can force the input processor to run before the other components.

Files. In order to facilitate testing, we recorded several hours of SAWD controller output in a disk file, which can be replayed at varying speeds. Since this data is from the hardware, it enabled us to confirm some operation before we had access to the hardware.

Input Processing

The input data from the SAWD controller via the RS-232 link are a series of floating-point numbers and one-byte flag values that reflect the current state of the hardware. These data are converted into the representation used by ARGES and a checksum is validated. If invalid, or if the display and expert system fall behind the input processor, the current data are discarded because the loss of a single data point is insignificant. If either the display or expert reasoner request data when the current value is invalid, they "sleep" until valid data becomes available.

The Expert Reasoner

The expert system operates in two modes: fault detection and fault diagnosis. In the fault detection mode, which occurs during normal SAWD operation, it monitors a few key parameters, or "health indicators", and the status and error-code values received from the controller. When these parameters indicate a potential fault in SAWD, the expert system enters fault diagnosis mode and begins to record data to determine the long-term data trends (if the system has not already shut down).

In the fault diagnosis mode incoming data are collected and stored, then analyzed and trended using linear least-squares fit for several SAWD cycles. Whenever sufficient data are collected, the result ing trend is inserted into the production system working memory and the rule base is invoked to diagnose and verify whether a fault has occurred.

When a fault is diagnosed, the rules invoke procedures to notify the user about the fault detected and provide recommendations. Display functions are also invoked to highlight the faulty component on the SAWD schematic. If the SAWD has not yet shut down, the data trends are extrapolated to find when they will cross thresholds and cause system shutdown. If no fault is recognized, a "default" rule is invoked to notify the operator that an undiagnosed fault exists. A tree of conclusions and antecedents, or inference net, is built as diagnosis proceeds to provide a means of constructing an explanation, discussed further below.

The Knowledge Representation

In ARGES, the expert knowledge is represented using HAPS (the Hierarchical, Augmentable Production System), an in-house developed production system similar to OPS5[Brownston84]. We chose HAPS because it was available at low cost, we had access to the source code so we could enhance it to meet any special requirements, and HAPS provided a reasonably flexible representation — it supports lists, frames, and FLAVOR instances as working memory elements; access to LISP functions on both the left and right hand sides of a rule; user-definable conflict resolution; and access to HAPS structures from the LISP environment.

Initially we used only simple lists for representations of working memory elements (domain knowledge facts), but this has been replaced with a frame hierarchy providing a fault taxonomy and description of the ECLSS. The diagnosis and fault recognition is performed by forward-chaining rules, with recommendations generated from the fault type using the frame hierarchy.

The User Interface

The user interface for ARGES performs several functions: a) it depicts graphically the operation of SAWD system to facilitate user understanding, b) it displays the conclusions reached by the expert system, c) it allows the user to examine the chain of reasoning used by the expert system. In order to study how an expert system could be integrated with the rest of a space station management system, the user interface simulates a more complete interface between a crew member and a more generalized fault management system. In particular, the fault management interface simulates other components of the overall space station management function and the status & warning display for the space station, although only that component dealing with the CO2 removal assembly is actually functional. By simulating these functions, we can demonstrate how an AI-based fault management system can enable a crewmember to detect and diagnose faults, perform temporary work-arounds, and take corrective actions with reduced knowledge of the Space Station systems and reduced ground support.

The interface uses two monitors, one color and one monchrome, and a mouse. It is based on the direct manipulation, object-oriented approach found in the Lisp machine and Smalltalk environment[Goldberg80, Symbolics85]. Text and objects displayed on both the monchrome and color screens are, to the extent possible, "mouse-sensitive". Inquiries concerning the represented systems, components, sensors, etc., are made by interacting with the displayed objects directly. In an operational system, the mouse would be replaced by a suitable zero-G device, such as a track ball.

The monchrome screen primarily displays textual information, such as error messages, and menus of possible corrective actions. Notifications of errors happen as highly-visible "pop-up" text boxes with an indication of the response time required by the user. A set of possible responses recommended by the expert system is an associated "pop-up" menu. In addition, significant events are recorded in a scrollable log. When the user requests, an additional window appears to display an explanation of the current conclusions.

To provide explanation, we make use of the inference net built during the fault diagnosis process. Associated with each working memory element is the description of the conclusion, which is easily converted to English text and displayed. Each displayed conclusion is mouse-sensitive, and clicking on it results in the addition (or removal) of its antecedent symptoms to the display. The user can traverse the inference net from the final conclusion back to the leaves of the net (i.e., trended data).

The color screen is the primary means of displaying information to the user. Under normal circumstances (before a fault has been recognized) a simple, hierarchical diagram of the space station systems is displayed. This "map" graphically provides the current status of various systems (off, on, warning, alarm, etc.). The user can click on a component box to see an expansion down to its subsystems with the status of each subsystem displayed. At the bottom of the hierarchy, the schematics of the assemblies are available, with the diagram dynamically updating to show the current status of each assembly. When a fault occurs, the appropriate subsystem boxes change state. In the assembly schematic, the icon of the faulted component also changes state, to graphically display the fault location isolated by the expert reasoner. As an aid to the user to help visualize the relationship of the faulty subsystem, the subsystem hierarchy is reproduced in the lower right corner, complete with status and possible other fault indications. This enables the user to always have an indication of overall status (or other problems) while dealing with a fault in one subsystem.

Current Status of System

ARGES has been implemented with a small set of faults on a Symbolics 3675 and an LMI Lambda. We have concentrated primarily on predictable faults, since this gives the additional capability of predicting a failure before it happens, as well as diagnosing a fault to a single component. We have tested it against the SAWD prototype at Marshall Space Flight Center for monitoring normal operations, but limitations on the testing of the hardware prevent us from introducing faults into the hardware to test the expert system. Currently, the system is undergoing an evaluation study of the expert system and user interface (see [Greitzer86]).

Conclusion and Future Directions

ARGES is a demonstration of AI technology applied to the problem of fault diagnosis and handling for manned space platforms. By applying AI/expert system technology, we can perform functions not possible with conventional controller software, such as: detection of degradations before they result in failure, providing greater flexibility in handling faults and modifying the fault software, and providing a higher level of interaction between the Space Station fault software and crewmember. The ultimate goal is to enable the crew to function more as managers and decision-makers, and less as interpreters of data and procedures manuals.

We are currently working to expand the simulation of the crew interface, so that more of the characteristics of an integrated "fault management" system can be evaluated. In addition, we would like to expand to a full set of faults for the entire atmosphere revitalization group. One of the limitations of the approach we have taken is that rule-based systems can only encode previously conceived faults. We would like to build a model-based reasoner, either quantitative, based on the simulation we use currently, or qualitative, to provide the deeper, causal reasoning necessary when the shallow rule-based approach is inadquate.

References

[Bailey86] Bailey, Pat, and Doehr, Brett, "Knowledge Acquisition and Rapid Prototyping of an Expert System: Dealing with "Real World" Problems", paper presented at the Conference on Artificial Intelligence for Space Applications, Nov. 13-14, 1986, Huntsville, Alabama.

[Boehm82] Boehm, Albert M., and Cusick, Robert J. "A Regenerable Solid Amine CO2 Concentrator for Space Station" Twelfth Intersociety Conference on Enviromental Systems, July 19-21, 1982, San Diego, CA.

[Browston84] Brownston, Lee and Robert Farrell, Elaine Kant, Nancy Martin, Programming Expert Systems in OPS5, Addison-Wesley, Reading, MA, 1984.

[Dickey84] Dickey, Frederick J., and Toussaint, Amy L. "ECECIS: An Application of Expert Systems to Manned Space Stations", Conference on Artificial Intelligence Applications Proceedings, IEEE, 1984.

[Goldberg80] Goldberg, Adele, Smalltalk-80, Addison-Wesley, Reading MA, 1980

[Greitzer86] Greitzer, Frank, "Intelligent Interface Design and Evaluation", paper presented at the Conference on Artificial Intelligence for Space Applications, Nov. 13-14, 1986, Huntsville, Alabama.

[Lance85] Lance, Nick and Malin, Jane T., "Development of a Prototype Expert System for Fault Diagnosis of a Regenerative Electrochemical CO2 Removal Subsystem", Ref. No. EC3/SR111, NASA/Johnson Space Center, Houston TX, May 13, 1985.

[Symbolics85] Symbolics, Inc., Symbolics Reference Manual, Vol. 7, Programming the User Interface, Cambridge, MA 1985.

[Yanosy85] Yanosy, James L., and Rowell, Lawrence F. "Utility of an Emulation and Simulation Computer Model for Air Revitalization System Hardware Design, Development, and Test" Fifteenth Intersociety Conference on Enviromental Systems, July 15-17, 1985, San Francisco, CA.